



Europäisches  
Patentamt

European  
Patent Office

Office européen  
des brevets

FCI/GB 2003 / 0 0 2 5 2 8

10/516952

Rec'd PATPTO 06 DEC 2004

REC'D 16 JUL 2003

WIPO PAT

Bescheinigung

Certificate

Attestation

Die angehefteten Unterla-  
gen stimmen mit der  
ursprünglich eingereichten  
Fassung der auf dem näch-  
sten Blatt bezeichneten  
europäischen Patentanmel-  
dung überein.

The attached documents  
are exact copies of the  
European patent application  
described on the following  
page, as originally filed.

Les documents fixés à  
cette attestation sont  
conformes à la version  
initialement déposée de  
la demande de brevet  
européen spécifiée à la  
page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

02254105.6

**PRIORITY  
DOCUMENT**

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

BEST AVAILABLE COPY

Der Präsident des Europäischen Patentamts;  
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets  
p.o.

R C van Dijk



Anmeldung Nr:  
Application no.: 02254105.6  
Demande no:

Anmeldetag:  
Date of filing: 12.06.02  
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

BRITISH TELECOMMUNICATIONS public limited  
company  
81 Newgate Street  
London EC1A 7AJ  
GRANDE BRETAGNE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:  
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.  
If no title is shown please refer to the description.  
Si aucun titre n'est indiqué se référer à la description.)

Video pre-processing

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s)  
revendiquée(s)  
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/  
Classification internationale des brevets:

H04N7/24

Am Anmeldetag benannte Vertragsstaaten/Contracting states designated at date of  
filing/Etats contractants désignées lors du dépôt:

AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR

## VIDEO PRE-PROCESSING

The present invention relates to a method and apparatus for pre-processing a video sequence and, in particular, to pre-processing a video sequence prior to transmitting  
5 it over a packet switched data network using a protocol such as MPG-4.

A video sequence comprises a sequential series of complete video frames intended to be displayed successively at a predetermined rate. Each frame comprises a matrix of pixels, each of which has, in the case of black and white video, an associated value  
10 indicating the brightness of the pixel, or, in the case of colour video, an associated triplet of values which together indicate both the brightness and the colour of each pixel.

A very common type of video sequence is one in which a single object (eg a person)  
15 moves in the foreground against a substantially static background. For example, a video teleconference in which two people sit in front of their respective computers with small cameras transmitting to one another video images of their faces (as well as audio information) will typically fall into this category of video sequence.

20 The present inventor has realised that in video sequences of this type, the most important aspect of the video sequence tends to be the object moving in the foreground. That is to say, inaccuracies, poor rendering or noise (ie a poor signal to noise ratio) tends to be more noticed by a viewer when it occurs within the foreground object.

25

Summary of the Invention

According to a first aspect of the present invention, there is provided a method of transmitting a video sequence across a non-ideal transmission channel in which a distinction is made between a foreground object part and a substantially static  
30 background part of each video frame within the sequence and the background part is transmitted at a lower quality than the object part.

The term quality as used herein includes the accuracy with which the videoed scene is portrayed, the resolution at which it is portrayed and the rate at which the depicted scene is updated, etc. Thus, according to the present invention, if there are limitations on the transmission channel through which a video sequence is to be sent  
5 (in terms of bandwidth, noise, loss of data, etc.) a viewer watching the received video sequence will tend to have a better viewing experience since better quality in the foreground object is achieved at the expense of the quality of the background, which, as mentioned above, is considered by the present inventor to contribute less strongly to the overall viewing experience of the viewer.

10

Preferably, the background part is transmitted at a lower update rate than the foreground object part. This is advantageous because the resultant reduction in quality of the background part is least likely to give rise to a reduced subjective viewing experience compared with other ways of reducing the bandwidth used to  
15 transmit the background part (and thus reducing the quality of the background part).

According to a second aspect of the present invention, there is provided a method of distinguishing between a foreground object part and a substantially static background part of each video frame within a video sequence, the method comprising the steps  
20 of:

dividing each video frame into a number of video blocks each of which comprises one or more pixels;

generating a mask frame in respect of each video frame, each mask frame having a mask block corresponding to each video block in each respective video  
25 frame; and

either setting each mask block to an object value, indicating that the corresponding video block in the corresponding video frame includes one or more pixels depicting a foreground object part, or setting each mask block to another value; wherein a mask boundary is defined in respect of each mask frame such that  
30 each mask block within the mask boundary is set to an object value whilst each mask block outside the boundary is set to another value, and wherein the mask boundary

position is varied between video frames according to a selected procedure, located in a computer program, which is stored in a computer-readable medium.

corresponding video frame, as determined by comparing the corresponding video frame with a substantially temporally adjacent video frame.

Preferably, each video block comprises sixteen by sixteen pixels. Video blocks of this  
5 size may hereinafter be referred to as macro-blocks.

According to one embodiment, each mask block can be set to either one of two possible values to indicate that the respective block is either an object block, in which case one or more of the pixels in the corresponding video block are considered  
10 to be object pixels, or a background block, in which case all of the one or more pixels in the corresponding video block are considered to be background pixels. In other embodiments, however, intermediate values may be used to indicate a degree of transparency of one or more pixels within the corresponding video block.

15 By stepped procedure, it is meant that the mask boundary may only be shrunk or expanded between successive mask frames by up to a predetermined maximum amount. Thus in one embodiment, each portion of the mask boundary can only be moved by up to two blocks away from the position of the corresponding portion of the mask frame in a preceding or successive mask frame.

20

According to a preferred embodiment of the present invention, the method of distinguishing between a foreground object part and a substantially static background part of each video frame within a video sequence comprises the steps of:

dividing each video frame into a number of video blocks each of which  
25 comprises one or more pixels;

generating a mask frame in respect of each video frame, each mask frame having a mask block corresponding to each video block in each respective video frame; and

either setting each mask block to an object value, indicating that the  
30 corresponding video block in the corresponding video frame includes one or more pixels depicting a foreground object part, or setting each mask block to another value; wherein the steps of generating each mask frame and setting each mask block comprise the following sub-steps:

a) selecting a first one of said video frames and setting it to be a past video frame;

b) generating a first mask frame corresponding to said first video frame, setting the first mask frame as a past mask frame and associating it with an initial  
5 mask boundary;

c) setting all mask blocks within the initial mask boundary to an object value;

d) selecting a subsequent one of said video frames which is substantially temporally adjacent to said past video frame and setting it as the current video frame;

e) comparing each pixel or group of pixels within said past video frame with  
10 each corresponding pixel or group of pixels within said current video frame and marking each pixel or group of pixels of said current video frame which differs from its corresponding pixel or group of pixels in said past video frame by more than a predetermined amount as a changed pixel or group of pixels;

f) determining where the most outwardly located changed pixel or group of  
15 pixels in each of a plurality of directed regions is located;

g) setting each portion of a current mask boundary associated with each respective directed region to a position which is either equal to the position of the corresponding portion of the current mask boundary or up to a predetermined maximum integer number of blocks away from the position of the corresponding  
20 portion of the current mask boundary towards the corresponding position of the most outwardly located changed video block in the respective directed region in said current video frame;

h) setting all mask blocks in the current mask frame which are located within the current mask boundary to an object value;

25 i) setting the current video frame and the current mask frame to be the past video frame and past mask frame respectively; and

j) repeating sub-steps d) to i) until a mask frame has been generated in respect of each of the video frames in the video sequence.

30.

Preferably, the first video frame which is selected is the video frame which is identified as the first frame of the video sequence and each subsequent video frame is identified as a frame which is temporally adjacent to the previous video frame.

th sequence, etc. However, in some circumstances, it may be advantageous to perform the processing backwards (ie starting from the last video frame to be displayed) or to skip intermediate frames (ie only processing every second, third fourth, etc. frame), or to divide the sequence into mini sequences, possibly  
5 overlapping, etc.

In order that the present invention may be better understood, embodiments thereof will now be described, by way of example only, with reference to the accompanying drawings in which:

10        Figure 1 is a block diagram of a video conference arrangement suitable for use with an embodiment of the present invention;

         Figure 2 is a schematic block diagram of a video frame divided into macro-blocks;

         Figure 3 is a schematic block diagram of a mask frame, illustrating the  
15        directions in which a video frame is scanned in the present embodiment to look for the outermost changed video block in each direction;

         Figure 4 is a schematic block diagram of a mask frame, illustrating the smallest size of mask which is permitted in the present embodiment;

         Figure 5 is a schematic diagram of a video frame illustrating how it is  
20        scanned from an upwards direction to locate the most outwardly located changed pixel in the upward direction;

         Figure 6 is a schematic diagram of a video frame illustrating how it is scanned from a downwards direction to locate the most outwardly located changed pixel in the downward direction;

25        Figure 7 is a schematic diagram of a video frame illustrating how, in the present invention, each video frame is conceptually divided into rows of one macro-block in height, for subsequent scanning from sideways directions;

         Figure 8 is a schematic diagram of a video frame illustrating how each row is scanned from the left to locate the most outwardly located changed pixel in the left  
30        direction in each row;

         Figure 9 is a schematic diagram of a video frame illustrating how each row is scanned from the right to locate the most outwardly located changed pixel in the right direction in each row;

Figure 10 is a schematic diagram of a video frame illustrating the result of all of the scans illustrated in Figures 5, 6, 8 and 9;

Figure 11 is a schematic illustration of a video frame, illustrating a high (single pixel) resolution mask formed by amalgamating the results of the scans shown in Figure 10; and

Figure 12 is a schematic illustration of a mask frame illustrating a low (macro-block) resolution mask, with the object image and high resolution mask superimposed thereon for illustrative purposes.

Figure 1 is a schematic illustration of a video conference arrangement. In this arrangement, two computer terminals 11, 12 are connected together via a Wide Area Network (WAN) 60 which permits packets of data to be transmitted between the computer terminals 11, 12 using the Internet Protocol, in combination with higher level transport mechanisms such as the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). Each computer terminal 11, 12 comprises a processor unit 21, 22, a microphone 31, 32 a monitor 41, 42 and a camera 51, 52.

In order to take part in a video conference with a second user 72, a first user 71 sits in front of the first terminal 11. The microphone 31 and video camera 51 of the first terminal 11 are trained on the first user 71 and record the first user's voice and image respectively. The second user 72 sits in front of the second terminal such that the microphone 32 and video camera 52 of the second computer terminal 12 are trained on the second user for recording his/her voice and image respectively. In addition to the first and second user's image, each video camera also records the backgrounds 81, 82 behind the first 71 and second 72 users respectively. The recorded analogue and video signals are digitised and processed for sending via the WAN 60 according to a suitable protocol such as MPEG-4.

In overview, the present embodiment aims to distinguish between a significant foreground object (ie users 71, 72) and relatively insignificant background objects (ie 81, 82), and to provide updated images only in respect of the significant foreground object. By doing this using a significant mask technique in which a mask is produced



which are not. The "shrinking" mask assumes that the object of interest will not be ring shaped (ie the part of the mask representing the significant foreground object does not contain holes representing the background). This is a reasonable assumption in most cases and especially in the video conference example illustrated in Figure 1 (in any event, the process of the present invention would still work with such foreground objects, just with slightly less than maximum efficiency). Additionally, the "shrinking" mask operates according to a step-by-step procedure, described in greater detail below, which makes the process reasonably robust against small errors or changes in background pixels from one frame to the next.

10

Referring now to Figure 2, each video frame comprises a given number of pixels, which, in the video frame illustrated in Figure 2 is  $96 \times 144 = 13,824$ . This is illustrated as forming  $6 \times 9 = 54$  macro-blocks 110, each of which contains  $16 \times 16 = 256$  pixels. Each pixel has an associated X and Y coordinate; in the example in Figure 2, the top left pixel has coordinates (0,0) and the bottom right pixel has coordinates (143,95). Thus the X-axis increases from left to right and the Y-axis increases from top to bottom. This is typical for pixel coordinates in video frames.

After some basic processing of the signals output from each camera 51, 52, each processor unit 21, 22 will produce a series of video frames intended to be displayed at a given rate (the refresh rate) to recreate the images recorded by the respective cameras. It is these video frames which provide the input to the process of the present embodiment which is described below in the form of a series of steps. The output of the process is the same video frames together with corresponding mask frames which are used to indicate which macro-blocks within each video frame need to be updated. A subsequent process then encodes and transmits over the WAN only those macro-blocks indicated by the mask frames as requiring updating. The subsequent process may, for example, operate in accordance with the well known MPEG-4 standard. On receiving the macro blocks requiring updating, the receiving computer terminal "reconstructs" a series of video frames which are displayed at the given refresh rate for viewing by the receiving user 72, 71.

### The Steps of the Process of the Present Embodiment

The following steps describe the process of the present embodiment:

#### Step 1

- 5 In the first step, the first video frame of the session is selected and a corresponding mask frame (having a macro-block resolution "shrinking" mask) is generated which, because it is the first frame, is automatically set with all of the mask blocks set to a value, which is hereinafter referred to as "transparent", which indicates that the corresponding macro-blocks within the corresponding video frame should be encoded
- 10 and transmitted. At the same time, a pixel resolution "shrinking" mask is generated; this is for use in steps 15 to 20 (see below); in step 1, as discussed below, this pixel resolution "shrinking" mask is also set to the outer perimeter of the frame size so as to include all possible pixels within the mask.

#### 15 Step 2

- In the second step, the next video frame is selected and a "subtraction operation" is performed on a pixel-by-pixel basis with respect to the preceding video frame (which will be the first frame when step 2 follows step 1, but not in subsequent iterations of the complete loop which loops round from step 2 to step 24 and back again), the
- 20 output of which is a differentiated frame in which pixels identified as non-differentiated are associated with a first value (eg 0) whilst pixels identified as differentiated pixels are associated with a second value (eg 1).

- Note that the "subtraction operation" is used to determine an approximate amount of
- 25 perceptual difference between the two pixels compared pixels (note the comparison may be done on a pixel by pixel basis or on a group of pixels by group of pixels basis – see below). The exact way in which this is done will depend on the type of coding used to code each individual pixel. In very simple cases each pixel is represented by a number which is correlated to a particular colour using a look-up table. In such
- 30 cases, the difference between the indexes used to look-up the corresponding colour tends to be related to the perceptual difference in the colour and so a simple

comparison of the difference between the two pixels can be used to determine the perceptual difference between the two pixels.

again a simple subtraction will indicate how perceptually different each pixel is from one another. In some colour representations, each of three different components (eg red, green & blue; or Luminance, u-value and v-value; etc.) has an associated number of bits with which to specify a particular value (which may vary from pixel to pixel if chromatic components are sampled less frequently than luminance components for example). In such a case, the subtraction operation may compare like components with like components (on a pixel by pixel basis) and then take an average to determine the overall perceptual difference of one pixel from another.

10 The aim of the subtraction process is to identify differentiated video areas. A threshold difference is used to avoid small noise causing pixels which are really substantially unchanged from being detected as changed pixels.

In the present embodiment, the process is optimised for colour video frames employing a pixel encoding protocol known as YUV(4:2:0). To avoid complications resulting from the lower levels of sampling of the colour characteristics compared to the luminance of each pixel, only the luminance values are compared between the current and previous video frames. Additionally, to minimise the problems caused by noise affecting only individual pixels (and not their neighbours), the comparison is done on the basis of a group of pixels (which in the present embodiment comprises a square of 2x2 pixels) and the average difference in luminosity per pixel across the group is calculated and compared with the threshold. In the present embodiment, this threshold is set to a default value of 20. This is deemed to be appropriate for YUV (4:2:0) pixel encoding in which the luminosity is encoded using 8 bits (ie to give a luminosity varying between 0 and 255). In an alternative embodiment, different defaults could be used to cater for different types of pixel encoding and the level could be user adjustable to fine tune the threshold depending upon the level of noise, the general brightness of the scene (with brighter scenes having a bigger threshold compared to darker scenes), etc.

30

As mentioned above, the present embodiment is optimised for processing video sequences containing a single, substantially centrally located, foreground object of interest. Thus the expectation is that the result of the subtraction process, which

identifies groups of differentiated pixels (where the differentiation is larger than the threshold value), will be a binary differentiated frame including a representation of the "dynamic object" in the centre of the frame, indicated by the marked differentiated groups of pixels.

5

### Step 3

A differentiated frame corresponding to the result of the subtraction operation performed in step 2 is prepared to be scanned from four different general directions, starting from the boundaries of the differentiated frame and heading towards the  
10 centre. The four different directions are from the top, the bottom, the left and the right, as shown in Figure 3.

### Step 4

A maximum distance to be covered in each of the scanning directions from each  
15 boundary of the differentiated frame towards the centre is set to avoid overlap and hence possible conflict between scans in opposed directions. These values could take on a range of different numbers depending on the nature of the video sequence. In the present embodiment, however, the following values are used, where X and Y represent the horizontal and vertical dimensions of the differentiated frame in terms  
20 of number of pixels:

$$\text{i) "Up" = } Y/2 - Y/8$$

$$\text{ii) "Down" = } Y/2 + Y/8$$

$$\text{iii) "Left" = } X/2 - X/8$$

$$\text{iv) "Right" = } X/2 + X/8$$

25

These values give rise to a "minimum" mask, because if, in any direction, no differentiated pixel or group of pixels is encountered before reaching the internal boundary set by the above values, then it is assumed that the outermost differentiated pixel in the respective direction is located at this internal boundary.

30

Figure 4 illustrates the resulting minimum mask set by the above quoted values used

IN THE PRESENT EMBODIMENT.

### Step 5

Having completed preparations for performing the scanning, the scanning commences and is described in detail below with reference to following steps 6 to 14.

#### 5 Step 6

This step involves scanning from the "Up" direction. Starting from the upper-left corner of the differentiated frame and moving towards the right, successive rows are scanned and each pixel is examined to see whether it is a differentiated (ie changed) or an undifferentiated (ie unchanged) pixel. As soon as a differentiated pixel is encountered, the scanning stops and the current row is set as the outer boundary of the "dynamic" object in the Up direction. The value of the current row is stored in a field which, in the present embodiment, is called *current\_up\_limit*. This step is illustrated in Figure 5.

#### 15 Step 7

This step is similar to Step 6 above except the scanning is from the "Down" direction, and scanning proceeds from the bottom right pixel, from right to left. As soon as a differentiated pixel is encountered, the scanning stops and the row in which it is detected is set as the outer boundary of the "dynamic" object in the Down direction. In the present embodiment, this value is stored in a field called *current\_down\_limit*. This step is illustrated in Figure 6.

### Step 8

Steps 8 to 13 are responsible for performing the scanning in the sideways directions. This is slightly different to the procedures described in steps 6 and 7, because the differentiated frame is first divided horizontally into "slices", each of which has a width equal to the width of a macro-block (ie 16 pixels). Thus each slice contains sixteen rows of pixels. This division into slices is schematically illustrated in Figure 7.

30

### Step 9

Steps 9 and 10 are responsible for performing the scanning in the "from the Left" direction. Each slice is considered in turn in step 9. If the slice is entirely above the

*current\_up\_limit* or entirely below the *current\_down\_limit*, then the slice is ignored and the next slice is selected. Otherwise, the slice is passed to step 10 for "from the Left" processing.

#### 5 Step 10

Each slice passed to step 10 for scanning is scanned, on a row-by-row basis, starting from the top left pixel of the slice and moving right until a differentiated pixel is encountered or the left (minimum) internal boundary is reached for that row. The X-coordinate of the first encountered differentiated pixel or the left internal boundary  
10 (as appropriate) is stored. This scanning from left to right is repeated for each of the sixteen rows of pixels within the slice to determine the left boundary of the "dynamic object". At the end of this process, all sixteen stored X-coordinates of each row within the slice are compared and the smallest value (ie the left-most differentiated pixel) is recorded in a field which, in the present invention, is called  
15 *current\_left\_limit\_slice\_n*, where n represents a number varying from 1 to N, where N is the number of slices in the differentiated frame (ie the left hand edge of the "dynamic object" as detected of the first slice is stored in *current\_left\_limit\_slice\_1*, and of the second slice in *current\_left\_limit\_slice\_2*, etc.). Note that alternative methods could be used to obtain this information; for example, each slice could be  
20 scanned on a mini-column by mini-column basis until a differentiated pixel is encountered. However, the presently employed method is preferred because one may, in a more sophisticated embodiment, utilise information about the shape of the left-hand boundary of the dynamic object to discount noisy differentiated pixels, which are much further to the left say than the average position of the left-hand  
25 boundary, and thus to get a better fix on the left-hand boundary of the actually sought for "dynamic object", etc.

#### Step 11

If all slices have now been considered in step 9, the process moves on to step 12.  
30 Otherwise, control passes back to step 9. The overall effect of steps 9, 10 and 11 is illustrated in Figure 8.

This step corresponds to steps 9 and 10, but for scanning from the Right. Thus each slice is considered. If the slice is entirely above the *current\_up\_limit* or entirely below the *current\_down\_limit*, then the slice is ignored and the next slice is selected. Otherwise, the slice is scanned, on a row-by-row basis, starting from the top right pixel of the slice and moving left until a differentiated pixel is encountered or the right (minimum) internal boundary is reached for that row. The X-coordinate of the first encountered differentiated pixel or the right internal boundary (as appropriate) is stored. This scanning from right to left is repeated for each of the sixteen rows of pixels within the slice to determine the right boundary of the "dynamic object". At the end of this process, all sixteen stored X-coordinates of each row within the slice are compared and the largest value (ie the right-most differentiated pixel) is recorded in a field which, in the present invention, is called *current\_right\_limit\_slice\_n*, where n represents a number varying from 1 to N.

#### 15 Step 13

Step 12 is repeated until all of the slices in the differentiated frame have been processed. Figure 9 schematically illustrates the result of steps 12 and 13.

#### Step 14

20 At this stage in the process, the following parameters have been calculated:

- a) "Up": *current\_up\_limit*
- b) "Down": *current\_down\_limit*
- c) "Left": *current\_left\_limit\_slice\_1*  
*current\_left\_limit\_slice\_2*

25

*current\_left\_limit\_slice\_N*

- d) "Right": *current\_right\_limit\_slice\_1*  
*current\_right\_limit\_slice\_2*

30

*current\_right\_limit\_slice\_N.*

These values are schematically illustrated in Figure 10.

Figure 11 illustrates a corresponding pixel resolution mask formed from the above set out values. This pixel resolution mask is assumed to envelope the "dynamic object" of interest.

In addition to making available the contents of the above set out fields relating to the current frame, corresponding values of the fields created during the preceding iteration of steps 2 to 24 are also made available for processing during the subsequent steps at this step (step 14). As can be seen from step 21 described below, the past values of these fields are stored in the following fields:

a) "Up": *past\_up\_limit*

b) "Down": *past\_down\_limit*

c) "Left": *past\_left\_limit\_slice\_1*

15                    *past\_left\_limit\_slice\_2*

.

.

*past\_left\_limit\_slice\_N*

d) "Right": *past\_right\_limit\_slice\_1*

20                    *past\_right\_limit\_slice\_2*

.

.

*past\_right\_limit\_slice\_N.*

25

Note that in the event that this step has been reached in the first iteration of steps 2 to 24, there will not have been created any previous values for the above set out fields. Therefore, each of the past fields is given an initial default setting which, in the present embodiment, corresponds to the outer boundary of the differentiated frame (see the discussion of setting the pixel resolution "shrinking" mask in step 1 above).



In steps 15 to 20, a pixel resolution shrinking mask is set in respect of the current differentiated frame according to the rules set out below, with respect to the past and current values discussed above.

- 5 In step 15, a new "Up" limit is determined and stored in the field *final\_up\_limit*.

Rule I) If the absolute value of the difference between the current and the past "Up" limits is less than or equal to 2, then set the final "Up" limit to equal the past "Up" limit. As will be seen below, this effectively results in the past "Up" limit remaining  
10 unchanged from one iteration of steps 2 to 24 to the next. This rule can be expressed in pseudo-code as:

IF {abs[(*current\_up\_limit*) - (*past\_up\_limit*)] ≤ 2}  
THEN *final\_up\_limit* = *past\_up\_limit*

15 Rule II) If the current "Up" limit is more than 2 pixels less than (ie higher than) the past "Up" limit then the final "Up" limit is set to equal the past "Up" limit minus a pre-determined "Up-move-out" amount which, in the present embodiment, is set to an initial default value of 1 pixel. As will be seen below, this effectively results in the past "Up" limit moving vertically upwards by the "Up-move-out" amount from one  
20 iteration to the next. This rule can be expressed in pseudo-code as:

IF { (*current\_up\_limit*) - (*past\_up\_limit*) < -2}  
THEN *final\_up\_limit* = *past\_up\_limit* - *up-move-out*  
WHERE *up-move-out* = 1, 2, 3, 4, 5, 6, 7, 8, 9 or 10 (default value is 1)

25 Rule III) If the current "Up" limit is more than 2 pixels more than (ie lower than) the past "Up" limit then the final "Up" limit is set to equal the past "Up" limit plus a predetermined "Up-move-in" amount which, in the present embodiment, is set to equal the "Up-move-out" amount. This rule can be expressed in pseudo-code as:

IF { (*current\_up\_limit*) - (*past\_up\_limit*) > 2}  
30 THEN *final\_up\_limit* = *past\_up\_limit* + *up-move-in*  
WHERE *up-move-in* = 1, 2, 3, 4, 5, 6, 7, 8, 9 or 10 (default value is 1)

### Step 16

This step corresponds to step 15 for the "Down" limit. Thus, expressed in pseudo-code terms only:-

- Rule I) IF  $\{ \text{abs}[(\text{current\_down\_limit}) - (\text{past\_down\_limit})] \leq 2 \}$   
 5 THEN  $\text{final\_down\_limit} = \text{past\_down\_limit}$
- Rule II) IF  $\{ (\text{current\_down\_limit}) - (\text{past\_down\_limit}) > 2 \}$   
 THEN  $\text{final\_down\_limit} = \text{past\_down\_limit} + \text{down-move-out}$   
 WHERE  $\text{down-move-out} = 1, 2, 3, 4, 5, 6, 7, 8, 9$  or 10 (default value is 1)  
 10
- Rule III) IF  $\{ (\text{current\_down\_limit}) - (\text{past\_down\_limit}) < -2 \}$   
 THEN  $\text{final\_up\_limit} = \text{past\_up\_limit} - \text{down-move-in}$   
 WHERE  $\text{down-move-out} = 1, 2, 3, 4, 5, 6, 7, 8, 9$  or 10 (default value is 1)
- 15 In the present embodiment,  
 $\text{down-move-out} = \text{up-move-out} = \text{down-move-in} = \text{up-move-in} = 1$

Step 17

This step corresponds to steps 15 and 16 for the Left direction. This is performed on a slice by slice basis. If a particular slice to be processed is either wholly above the final Up limit calculated in step 15 or wholly below the final Down limit calculated in  
 5 step 16, then the slice is ignored and the process proceeds to step 18; otherwise one of the following rules (again expressed only in pseudo-code for the sake of brevity) is exercised as appropriate:

Rule I) IF {  $\text{abs}[(\text{current\_left\_limit\_slice\_n}) - (\text{past\_left\_limit\_slice\_n})] \leq 2$  }  
 10 THEN  $\text{final\_left\_limit\_slice\_n} = \text{past\_left\_limit\_slice\_n}$

Rule II) IF {  $(\text{current\_left\_limit\_slice\_n}) - (\text{past\_left\_limit\_slice\_n}) < -2$  }  
 THEN  $\text{final\_left\_limit\_slice\_n} = \text{past\_left\_limit\_slice\_n} - \text{left-move-out}$   
 WHERE  $\text{left-move-out} = 1, 2, 3, 4, 5, 6, 7, 8, 9$  or 10 (default value is 1)  
 15

Rule III) IF {  $(\text{current\_left\_limit\_slice\_n}) - (\text{past\_left\_limit\_slice\_n}) > 2$  }  
 THEN  $\text{final\_left\_limit\_slice\_n} = \text{past\_left\_limit\_slice\_n} + \text{left-move-in}$   
 WHERE  $\text{left-move-in} = 1, 2, 3, 4, 5, 6, 7, 8, 9$  or 10 (default value is 5 initially)

20 Note that in the present embodiment,  $\text{left-move-in} \neq \text{left-move-out} = \text{up-move-out}$ . This has the effect that in the sideways directions, the shrinking mask is able to shrink more quickly than it can expand. This assists the shrinking mask, and thus ultimately the mask frame, to encompass only the dynamic object of interest in a relatively short number of iterations thus reducing the amount of bandwidth required  
 25 to be sent to transmit the video information. However, once the mask has shrunk to fit the dynamic object, it is no longer necessary to leave the move-in value so high; step 21 below deals with this in the present embodiment.

Step 18

30 The process is returned to step 17 until all of the slices have been dealt with by step 17. Thereafter, the process moves on to step 19.

Step 19

This step determines the final right limits and corresponds to step 17. As before, this step is performed on a slice by slice basis. If a particular slice to be processed is either wholly above the final Up limit calculated in step 15 or wholly below the final Down limit calculated in step 16, then the slice is ignored and the process proceeds  
 5 to step 20; otherwise one of the following rules (again expressed only in pseudo-code for the sake of brevity) is exercised as appropriate:

Rule I) IF {  $\text{abs}[(\text{current\_right\_limit\_slice\_n}) - (\text{past\_right\_limit\_slice\_n})] \leq 2$  }  
 THEN  $\text{final\_right\_limit\_slice\_n} = \text{past\_right\_limit\_slice\_n}$

10

Rule II) IF {  $(\text{current\_right\_limit\_slice\_n}) - (\text{past\_right\_limit\_slice\_n}) > 2$  }  
 THEN  $\text{final\_right\_limit\_slice\_n} = \text{past\_right\_limit\_slice\_n} + \text{left-move-out}$   
 WHERE  $\text{right-move-out} = 1, 2, 3, 4, 5, 6, 7, 8, 9$  or 10 (default value is 1)

15 Rule III) IF {  $(\text{current\_right\_limit\_slice\_n}) - (\text{past\_right\_limit\_slice\_n}) < -2$  }  
 THEN  $\text{final\_right\_limit\_slice\_n} = \text{past\_right\_limit\_slice\_n} - \text{right-move-in}$   
 WHERE  $\text{right-move-in} = 1, 2, 3, 4, 5, 6, 7, 8, 9$  or 10 (default value is 5 initially)

Again, in the present embodiment,  $\text{right-move-in} \neq \text{right-move-out} = \text{up-move-out}$ .

20

#### Step 20

The process is returned to step 19 until all of the slices have been dealt with by step 19. Thereafter, the process moves on to step 21.

#### 25 Step 21

As mentioned above with reference to step 17, once the mask has shrunk to fit around the dynamic object, it is no longer necessary to have the left and right move-in values maintained at a high level. Thus in this step it is determined, in the present embodiment, whether more than a predetermined number of iterations has taken  
 30 place (which, in the present embodiment is set at 50 iterations) and if so the value for the left and right move-in variables is reduced from 5 to 1. Additionally, in the present embodiment, at this time, the value for the left, right, up and down move-out variables is increased from 1 to 5 at this point to enable the process to react more quickly in the event of an "expansion" of the dynamic object (ie the "region of interest") because, in this case a significant amount of information may be lost if

interest, the process is able to react faster when one or more of the boundaries of the object are moving towards one or more of the edges of the picture.

- Of course, in an alternative embodiment, one or more of the move-out variables could be set at a higher value than the or each corresponding move-in variable right from the start, or after some other predetermined point in the process. Also, instead of simply counting up to a predetermined number of iterations before changing the values of one or more of the move-out or move-in variables, some other indicator could be used to determine when such a change should occur. For example, a reduction in move-in values could be made after a certain (small) number of iterations have happened in which the final pixel resolution limits have remained unchanged from one iteration to the next, etc.

#### Step 22

- In this step, the "final" pixel resolution limit fields are used to generate macro-block resolution boundaries from which a mask frame corresponding to the current video frame is created in the next step. To do this, an inter macro-block boundary is chosen in respect of each of the final limit fields (ie *final\_up\_limit*, *final\_down\_limit*, *final\_left\_limit\_slice\_1*, *final\_left\_limit\_slice\_2*, ..., *final\_left\_limit\_slice\_N*, *final\_right\_limit\_slice\_1*, *final\_right\_limit\_slice\_2*, ..., *final\_right\_limit\_slice\_N*) except that, as per normal, slices either wholly above the final up limit or wholly below the final down limit are ignored. Each inter macro-block boundary is selected by moving away from the centre of the frame from the respective limit value in the respective direction until an inter macro-block boundary is encountered, unless the pixel resolution limit already lies on an inter macro block boundary. The inter macro-block boundaries are inclusive such that in the up direction the top row of pixels in each macro-block represents an inter macro-block boundary, in the right direction the rightmost column of pixels in each macro-block represents the inter macro-block boundary, etc.

30

#### Step 23

- Having selected a corresponding macro-block boundary in respect of each final pixel resolution limit, a mask frame is generated using the selected macro-block boundaries such that all macro-blocks contained within the selected macro-block boundaries are selected macro-blocks. This is illustrated in Figure 12.

35

#### Step 24

- This is the final step in the main iteration. Provided there is at least one more video frame requiring a corresponding mask frame to be generated prior to transmitting the video information, the past limit fields are reset to hold the final limit values (the pixel resolution ones, not the macro-block resolution boundaries used to generate the actual mask frame) (ie set *past\_up\_limit* = *final\_up\_limit*; etc.). The current video frame is set as the past video frame. The next video frame requiring a mask frame is selected and the process is returned to step 2 to repeat the iteration from steps 2 to 24 with the new video frame.

45

**Further Processing**

As mentioned above, the output from the above process will be a series of the originally input video frames together with corresponding mask frames. These can then be provided to a suitable video encoder such as any commercially available  
5 video encoder which operates in accordance with a standard such as MPEG-4 which permits objects to be encoded and transmitted without the surroundings.

At the receiver end, a decoder is used which keeps the decoded picture from the previous frame and adds the "new decoded frame" on top of that. In this way, the  
10 entire frame will be displayed to the viewer, without him or her noticing that only part of that frame was actually encoded and sent.

## CLAIMS

1. A method of transmitting a video sequence across a non-ideal transmission channel in which a distinction is made between a foreground object part and a substantially static background part of each video frame within the sequence and the background part is transmitted at a lower quality than the object part.
2. A method according to claim 1 wherein the background part is transmitted at a lower update rate than the foreground object part.
3. A method of distinguishing between a foreground object part and a substantially static background part of each video frame within a video sequence, the method comprising the steps of:
  - dividing each video frame into a number of video blocks each of which comprises one or more pixels;
  - generating a mask frame in respect of each video frame, each mask frame having a mask block corresponding to each video block in the respective video frame; and
  - either setting each mask block to an object value, indicating that the corresponding video block in the corresponding video frame includes one or more pixels depicting a foreground object part, or setting each mask block to another value; wherein a mask boundary is defined in respect of each mask frame such that each mask block within the mask boundary is set to an object value whilst each mask block outside the boundary is set to another value, and wherein the mask boundary position is varied between mask frames, according to a stepped procedure, towards a position in which the boundary just encompasses the foreground object part, of the corresponding video frame, as determined by comparing the corresponding video frame with a substantially temporally adjacent video frame.
4. A method according to claim 3 wherein each video block comprises sixteen by sixteen pixels.

5. A method according to claim 3 or 4 wherein the mask boundary may only be shrunk or expanded between successive mask frames by up to a predetermined maximum amount.
- 5 6. A method according to claim 3, 4 or 5 wherein the steps of generating each mask frame and setting each mask block comprise the following sub-steps:
  - a) selecting a first one of said video frames and setting it to be a past video frame;
  - b) generating a first mask frame corresponding to said first video frame,
  - 10 setting the first mask frame as a past mask frame and associating it with an initial mask boundary;
  - c) setting all mask blocks within the initial mask boundary to an object value;
  - d) selecting a subsequent one of said video frames which is substantially temporally adjacent to said past video frame and setting it as the current video frame;
  - 15 e) comparing each pixel or group of pixels within said past video frame with each corresponding pixel or group of pixels within said current video frame and marking each pixel or group of pixels of said current video frame which differs from its corresponding pixel or group of pixels in said past video frame by more than a predetermined amount as a changed pixel or group of pixels;
  - 20 f) determining where the most outwardly located changed pixel or group of pixels in each of a plurality of directed regions is located;
  - g) setting each portion of a current mask boundary associated with each respective directed region to a position which is either equal to the position of the corresponding portion of the past mask boundary or up to a predetermined maximum
  - 25 integer number of blocks away from the position of the corresponding portion of the past mask boundary towards the corresponding position of the most outwardly located changed video block in the respective directed region in said current video frame;
  - h) setting all mask blocks in the current mask frame which are located within
  - 30 the current mask boundary to an object value;
  - i) setting the current video frame and the current mask frame to be the past video frame and past mask frame respectively and



j) repeating sub-steps d) to i) until a mask frame has been generated in respect of each of the video frames in the video sequence.

7. A carrier medium carrying computer processor implementable instructions for  
5 causing a computer processor to carry out the method of any one of the preceding claims during implementation of the instructions.

8. A video-conference device including one or more components arranged to perform the method of any of claims 1 to 6.

10

9. A device for transmitting a video sequence across a non-ideal transmission channel, the device including means for distinguishing between a foreground object part and a substantially static background part of each video frame within the sequence and means for transmitting the background part at a lower quality than the  
15 object part.

## ABSTRACT

## VIDEO PROCESSING

5 A method of distinguishing between a foreground object part and a substantially static background part of each video frame within a video sequence. The method comprises the steps of:

dividing each video frame into a number of video blocks each of which comprises one or more pixels;

10 generating a mask frame in respect of each video frame, each mask frame having a mask block corresponding to each video block in each respective video frame; and

either setting each mask block to either an object value, indicating that the corresponding video block in the corresponding video frame includes one or more  
15 pixels depicting a foreground object part, or to another value.

The steps of generating each mask frame and setting each mask block comprise the following sub-steps:

a) selecting a first one of said video frames and setting it to be a past video  
20 frame;

b) generating a first mask frame corresponding to said first video frame, setting the first mask frame as a past mask frame and associating it with an initial mask boundary;

c) setting all mask blocks within the initial mask boundary to an object value;

25 d) selecting a subsequent one of said video frames which is substantially temporally adjacent to said past video frame and setting it as the current video frame;

e) comparing each pixel or group of pixels within said past video frame with each corresponding pixel or group of pixels within said current video frame and marking each pixel or group of pixels of said current video frame which differs from  
30 its corresponding pixel or group of pixels in said past video frame by more than a predetermined amount as a changed pixel or group of pixels;

f) determining where the most outwardly located changed pixel or group of pixels in each of a plurality of directed regions is located;

g) setting each portion of a current mask boundary associated with each respective directed region to a position which is either equal to the position of the corresponding portion of the past mask boundary or up to a predetermined maximum integer number of blocks away from the position of the corresponding portion of the past mask boundary towards the corresponding position of the most outwardly located changed video block in the respective directed region in said current video frame;

h) setting all mask blocks in the current mask frame which are located within the current mask boundary to an object value;

10 i) setting the current video frame and the current mask frame to be the past video frame and past mask frame respectively; and

j) repeating sub-steps d) to i) until a mask frame has been generated in respect of each of the video frames in the video sequence.

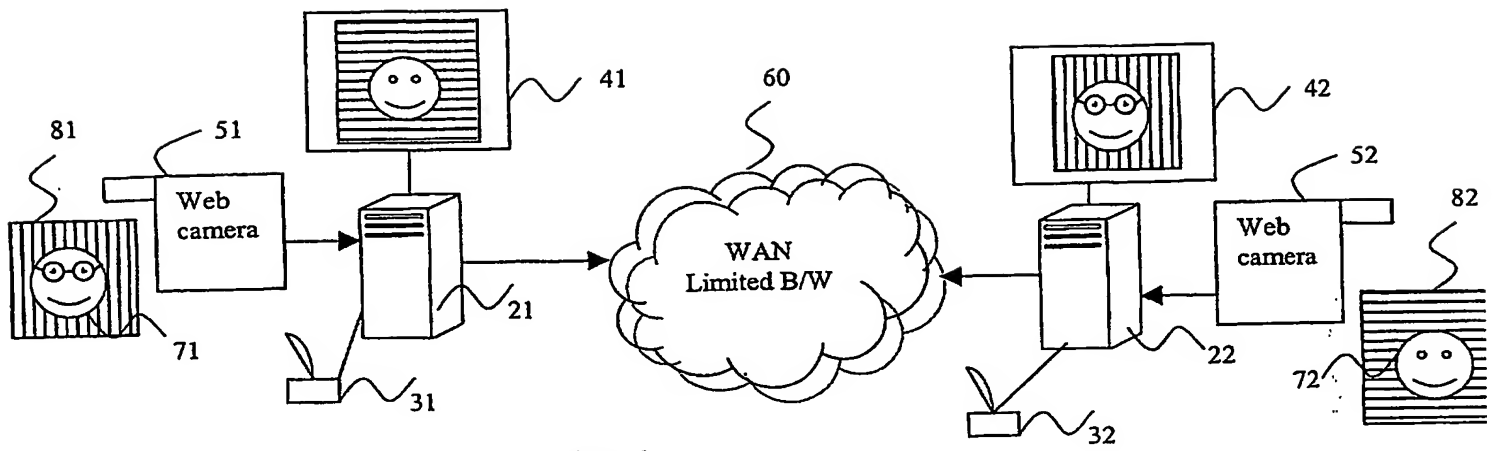


Figure 1

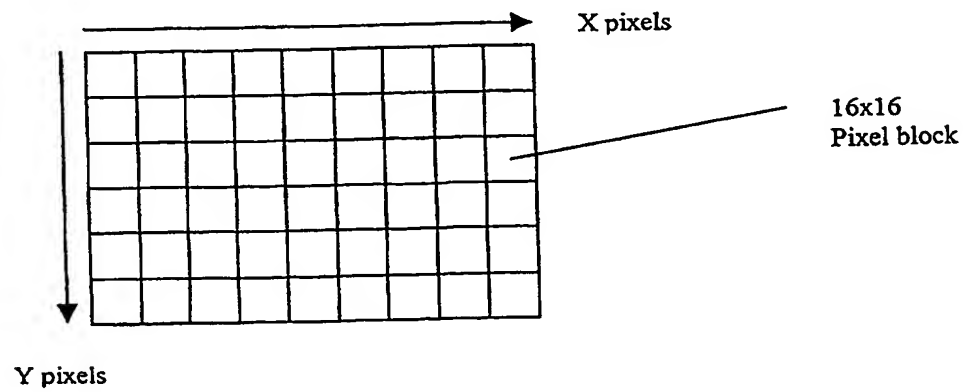


Figure 2

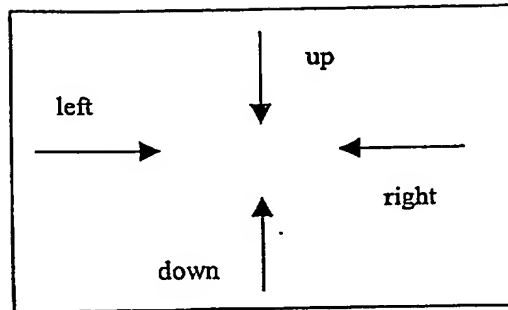


Figure 3

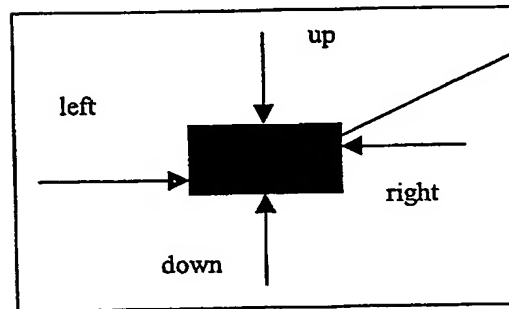


Figure 4

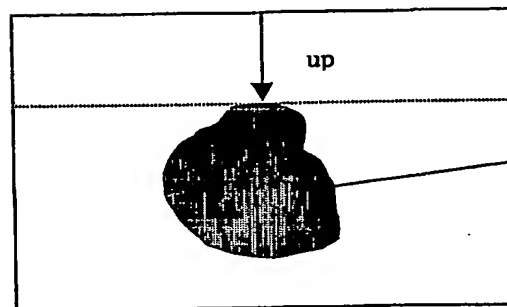


Figure 5

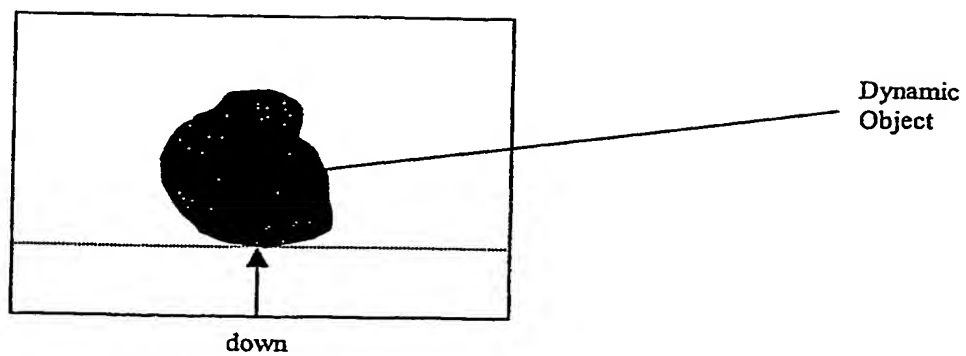


Figure 6

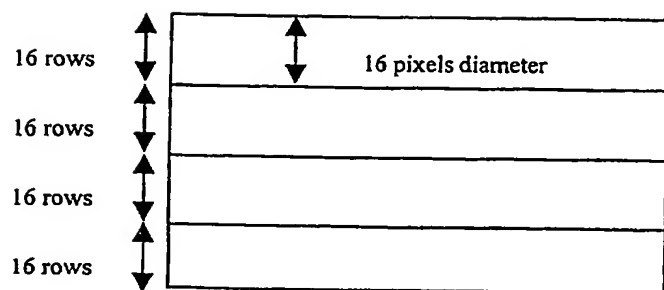


Figure 7

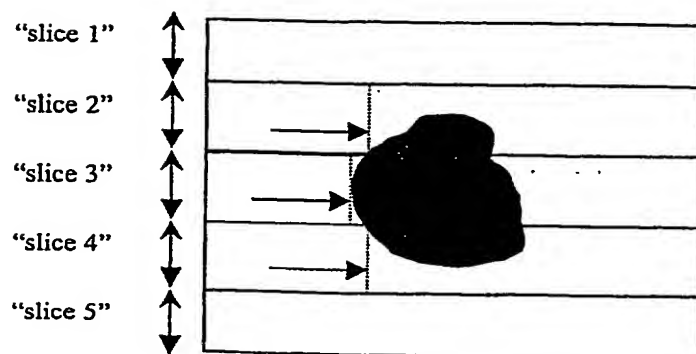


Figure 8

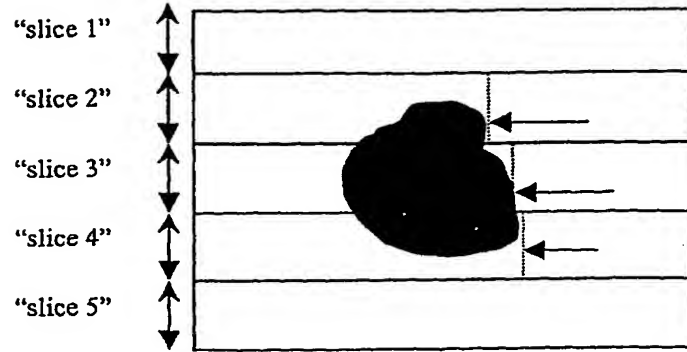


Figure 9

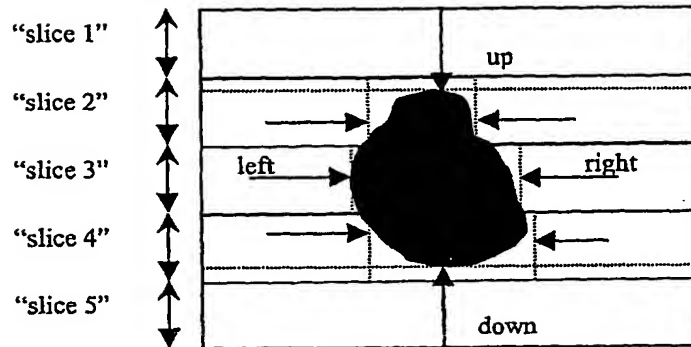


Figure .10

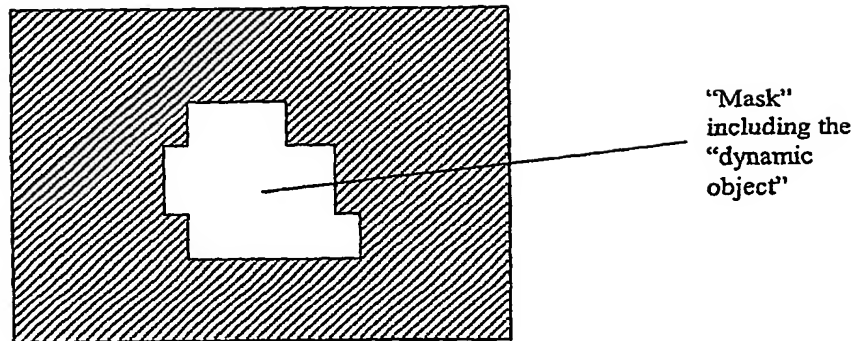


Figure 11

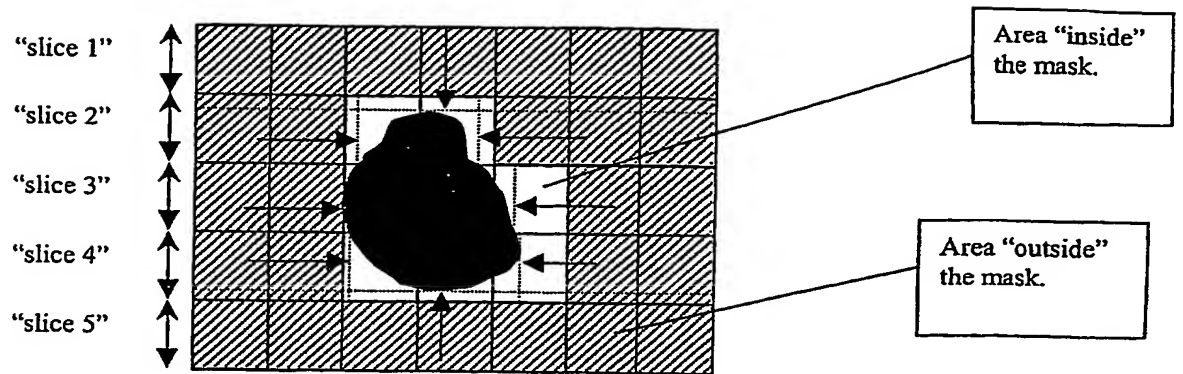


Figure 12



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**